



EESTI MAAÜLIKOOL
Tehnikainstituut

Klaid Hännile

**RASPBERRY PI-L PÕHINEV ÕPPEVAHEND
VOOLURINGIDE UURIMISEKS**

RASPBERRY PI BASED DEVICE TO STUDY ELECTRICAL
CIRCUITS

Bakalaureusetöö
Tehnika ja tehnoloogia õppekava

Juhendaja: doktorant Heino Pihlap, *MSc*

Tartu 2021

Eesti Maaülikool		Bakalaureusetöö	
Kreutzwaldi 1, Tartu 51006			
Autor: Klaid Hännile		Õppekava: Tehnika ja tehnoloogia	
Pealkiri: Raspberry PI-l põhinev õppevahend vooluringide uurimiseks			
Lehekülgi: 41	Jooniseid: 15	Tabeleid: 1	Lisasid: 2
<p>Osakond : Energiakasutuse õppetool</p> <p>ETIS-e teadusvaldkond ja CERC S-i kood: 4. Loodusteadused ja tehnika</p> <p>4.17. Energeetikaalused uuringud</p> <p>T140 Energeetika</p> <p>Juhendaja: Heino Pihlap</p> <p>Kaitsmiskoht ja -aasta: Tartu, 2021</p>			
<p>Elektrotehnika õppimisel on oluline, et kindlasti oleksid selged just põhitõed. See tähendab, et elektrotehnika huviline võiks teada, mida tähendab oomiseadus, kuidas elektrivool vooluringi erinevates harudes liigub ning mis juhtub, kui tarbijad on rööp-, jada- või segaühenduses.</p> <p>Et lihtsustada vooluringidest arusaamist, on töö autor võtnud eesmärgiks koostada seadeldis, millega oleks võimalik uurida ja õppida vooluringiseaduseid. Seadeldis võimaldab kasutajal muuta vooluringi olemust, kuvades ekraanil vooluringi pinged, voolud ja kogutakistuse.</p> <p>Koostati vooluring, mis koosneb üheksast takistist ja kolmeteistkümnest lülitist, millede lülitamisel on võimalik takisteid vooluringist eemaldada lisada. Seadme valmistamiseks kasutati Raspberry PI-d ning kahte MCP3008 analoog- digitaalmuundurit, milledega on võimalik mõõta pingelangusid takistitel.</p> <p>Koostati ja prinditi 3D mudel, kuhu saab kasutatavad komponendid paigutada. Kirjutati kood, mis kuvaks väärtused ekraanil. Viimaseks paigutati seadmed 3D prinditud karpi, joodeti kokku vooluring ning katsetati, kas seadeldis kuvab ekraanil korrektsed tulemused.</p>			
Märksõnad: Raspberry PI, vooluring, analoog- digitaalmuundur,			

Estonian University of Life Sciences Kreutzwaldi 1, Tartu 51014		Bachelor's Thesis	
Author: Klaid Hännile		Curriculum: Engineering	
Title: Raspberry PI Based Device to Study Electrical Circuits			
Pages: 41	Figures: 15	Tables: 1	Appendixes: 2
Department / Chair: Chair of Energy Application Engineering Field of research and (CERC S) code: 4. Natural Sciences and Engineering 4.17. Energetic Research T140 Energy research Supervisor: Heino Pihlap Place and date: Tartu, 2021			
<p>In Electrical studies, it is important that the fundamental laws are clear. It means, that one studying Electrics, should know what is Ohm's law, how current is divided in different parts of a circuit and what is the difference between parallel, series or both circuits.</p> <p>To make it easier to understand circuits, the author of this thesis has an aim to devise and assemble a device that would help to study laws of electrical circuits. The device would allow the user to modify the circuit and display circuits voltages, currents and resistance on a screen.</p> <p>A circuit was created that consists of nine resistors and thirteen switches, which allow the user to add or remove resistors from the circuit. Raspberry PI was used as the processing unit for the device and two MCP3008 analogue- digitalconverters, to measure voltagedrops on each resistor.</p> <p>3D model was modeled and printed, where it would be possible to place all components. Code was programmed, that would display information on a screen. Lastly the components were secured inside the 3D printed box and the circuit was soldered together and the device was tested to see, if displayed information is accurate.</p>			
Keywords: Raspberry PI, electrical circuit, analogue- digitalconverter			

SISUKORD

SISSEJUHATUS.....	5
1. VOOLURINGI SEADUSTE UURIMINE.....	6
1.1. Jada- ja rööpvooluringi eripärad.....	6
1.2. Ohmiseadus, Kirchhoffi seadused.....	7
1.3. Seadeldis vooluringiseaduste uurimiseks.....	7
2. SEADME KOMPONENTIDE VALIK	9
2.1. Vooluringi koostamine	9
2.2. Arvutusriistvara.....	10
2.3. Analoo- digitaalsignaalmuundur MCP 3008.....	11
2.4. Takistid.....	12
2.5. Lülitid	13
3. SEADME VALMISTAMINE.....	15
3.1. Raspberry PI ülesseadmine	15
3.1.1. Raspberry PI esmased sätted.....	15
3.1.2. SPI liidese lubamine	16
3.2. Analoo- digitaalmuunduri ühendamine.....	17
3.2.1. MCP 3008 füüsiline ühendamine.....	17
3.2.2. Analoo- digitaalmuunduri installeerimine Raspberry-s.....	19
3.3. 3D prinditava detaili loomine.....	20
4. TARKVARALISE KASUTAJALIIDESE LOOMINE.....	23
4.1. <i>Python</i> programmeerimiskeel.....	23
4.2. Tkinter	23
4.3. Ekraanil kuvatavate väärtuste leidmine.....	25
4.3.1. Pinge mõõtmine	25
4.3.2. Voolu tugevuse ja kogutakistuse arvutamine.....	26
5. SEADME MONTEERIMINE JA KASUTAMINE.....	28
5.1. Seadme kokkupanek.....	28
5.2. Seadme kasutamine	30
KOKKUVÕTE	32
KASUTATUD KIRJANDUS.....	33
LISAD.....	35

SISSEJUHATUS

Elektrotehnika õppimisel on oluline, et kindlasti oleksid selged just põhitõed. See tähendab, et elektrotehnika huviline võiks teada, mida tähendab oomiseadus, kuidas elektrivool vooluringi erinevates harudes liigub ning mis juhtub, kui tarbijad on rööp-, jada- või segaühenduses.

Kogemus on näidanud, et õppimine ning olulise informatsiooni meeldejäämine toimub just siis kõige paremini, kui inimene saab oma kätega katsuda ja uurida. Selle tarbeks loob töö autor seadme, mis võimaldab inimesel tarbijaid vooluringis sisse ja välja lülitada, muutes sellega kogu vooluringi olemust. Samal ajal on võimalik jälgida ekraanilt toimuvat – milline on vooluringi harudes olev vool, missugune pinge mõjub kõikidele tarbijatele ning kui suur on vooluringi kogutakistus.

Töö eesmärgiks on luua seadeldis, mis võimaldaks kuvada ekraanil vooluringi parameetreid ning samal ajal oleks võimalik vooluringi manipuleerida, et näha, kuidas parameetrid muutuvad.

Töö on jaotatud viieks osaks. Esimeses tutvustatakse vooluringi seaduseid ning kuidas neid oleks võimalik mõne seadeldisega uurida. Teises peatükis antakse ülevaade seadme komponentide valikust. Kolmandas peatükis tutvustatakse Raspberry PI ning analoog-digitaalsignaali muunduri ülesseadmist ja seadmele 3D hoidiku valmistamist. Neljas peatükk kirjeldab tarkvaralise kasutajaliidese loomist ning viimane peatükk annab valmisproduktist ülevaate.

Autor tänab enda juhendajat, Heino Pihlapit, heade nõuannete ja abi eest 3D mudeli printimisel.

1. VOOLURINGI SEADUSTE UURIMINE

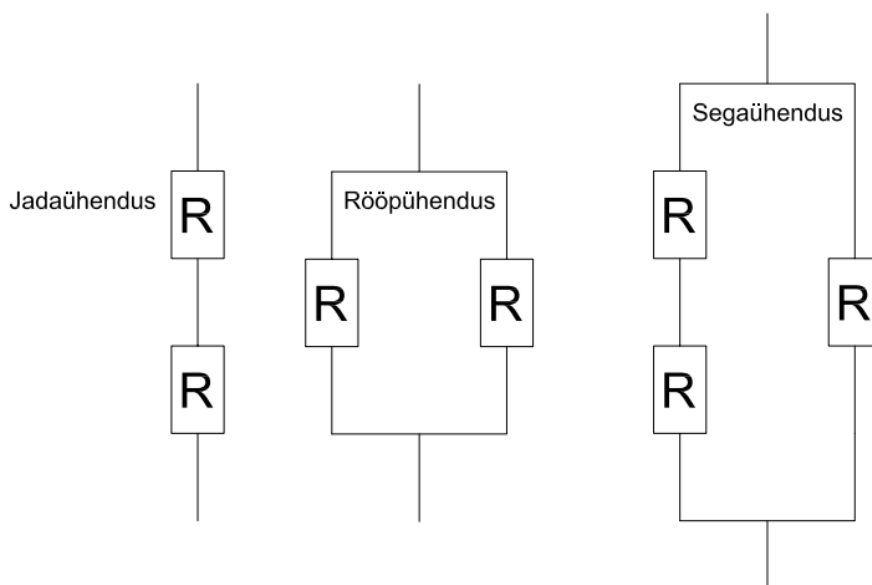
1.1. Jada- ja rööpvooluringi eripärad

Elektriskeemide puhul eristatakse kolme erinevatelektrikomponentide ühendusviisi:

1. Jadaühendus;
2. Rööpühendus;
3. Segaühendus.

Jadaühenduse puhul on kõik seadmed ühendatud omavahel jadamisi ehk järjest, rööpühenduses seadmed paralleelselt ehk vool on jaotatud mitmesse harusse. Segaühenduses esineb nii jada- kui ka rööpühendus. [1]

Jada- ja rööpvooluringi suurimad eripärad on pinge ja voolu jagunemine ning kogutakistus. Jadamisi ühendatud komponentidel on voolutugevus sama, kuid pinge erinev ning takistus on iga tarbija takistuste kogusumma. Rööpühenduses komponentidel on pinge sama, kuid vool erinev ja takistus on tarbijate takistuste pöördväärtuste summa. [1]



Joonis 1.1. Jada-, rööp- ja segaühenduse näide

1.2. Ohmiseadus, Kirchhoffi seadused

Alalisvooluringis on pinge, voolutugevus ja takistus omavahel seoses, mille avastas G.S. Ohm 19. sajandil. Ohm leidis, et vooluringi vool on pinge ja takistuse jagatis ehk

$$I = \frac{U}{R}.$$

Tema auks nimetati käesolev seadus Ohmi seaduseks. See tähendab, et kui vooluringis on teada kaks väärtust, saab alati leida kolmanda, mis on aluseks kõiksugu vooluringide koostamiseks. [1]

Peale Ohmi seaduse avastamist leidis G.R. Kirchhoff, et vooluringi kõikides harudes peab pingelang olema sama suur, kui on elektriahela elektromotoorjõudude summa. See tähendab, et vooluringi pingelangude ning elektromotoorjõudude kogusummade kokkuliitmisel peab vastuseks olema null. [2]

Kirchhoffi pingeseaduse tundmine võimaldab leida vooluringis kindlaid pingete väärtuseid, mida ainult Ohmi seaduse kasutamisega ei ole võimalik leida. [1]

Kirchhoff sõnastas ka teise seaduse, mis ütleb, et harusse sisenevate voolutugevuste ning harust väljuvate voolutugevuste summa peab olema null. See tähendab, et on võimalik leida tundmatu haru voolutugevus, kui on teada ülejäänud harude voolutugevus. [2]

1.3. Seadeldis vooluringiseaduste uurimiseks

Teadmisest, kuidas vool, pinge ja takistus käituvad, võib esialgu olla raske arusaada, vaadates ainult tehteid paberil. Selle asemel võiks inimesel olla võimalus omal käel vooluringi modifitseerida ning samaaegselt näha, mis muutub ülejäänud vooluringiparameetritega, et teadmine oleks paremini mõistetav.

Kui inimesel oleks võimalik muuta vooluringis näiteks harude arvu ning koheselt näha, kuidas ülejäänud harudes parameetrid sellega muutuvad, saaks kasutaja veenduda, et harusse sisenevad voolud peavad olema võrdsed sellest väljuvate voolude summaga.

Pingelangude uurimiseks oleks võimalik selline süsteem, kus seadeldise kasutaja eemaldaks vooluringist tarbijaid, mille tõttu ülejäänud tarbijate pinge tõuseks pidevalt, sest pingelangude algebraline summa peab alati olema võrdne toitepingega.

Vooluringide paremaks tutvustamiseks oleks vaja luua seadeldis, millel oleksid kõik eelnevalt mainitud funktsioonid. Kasutaja saaks muuta vooluringi parameetreid ning tulemused oleksid selgelt näha koheselt peale vooluringi manipuleerimist.

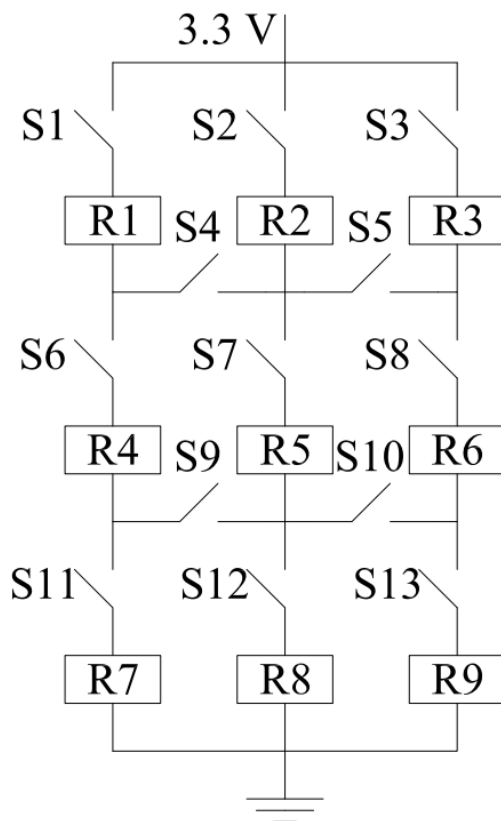
Tulemuste kuvamiseks oleks võimalik kasutada näiteks arvutiekraani, kus oleks selgelt näha vooluring ning selle hetkeparameetrid, mis tähendab, et seadeldise südameks peab olema mõni arvutiprotsessor, mis suudab pidevalt monitoorida vooluringi ning arvutada ekraanil kuvatavaid parameetreid.

2. SEADME KOMPONENTIDE VALIK

2.1. Vooluringi koostamine

Projektis kasutatakse vooluringi, mis on ühendatud 9 takistiga 13 lülitiga. Vooluringi peab võimaldama kasutajale näidata, kuidas vool, pinge ning takistus muutuvad sõltuvalt jada- ja paralleelühendustest, et tutvustada oomiseadust ning näidata, kuidas Kirchhoffi seadused töötavad.

Selle jaoks on vaja luua vooluring, kus on mitmed tarbijad, mis on omavahel ühendatud nii jadamisi kui ka rööbiti. Kasutada tuleks ka lüliteid või releesid, mis vooluringi erinevates harudes paikneksid. See annaks võimaluse vooluringis suurendada või vähendada harude arvu, kust kaudu vool saab liikuda.



Joonis 2.1. Vooluringi skeem

2.2. Arvutusriistvara.

Seadeldis, mille abil vooluringe uurida saaks peab sisaldama mingit sorti arvutit, sest muidu ei oleks võimalik kuvada ekraanil vooluringis toimuvat. Lisaks võiks olla tegemist seadmega, mis oleks kaasaskantav. Näiteks lauaarvuti või rüperaali kasutamine oleks välistatud, sest seade muutuks liialt suureks. Arvutituumaks võiks valida näiteks Arduino või Raspberry PI.

Arduino puhul on tegemist pisikese mikrokontrolleriga, millele kasutaja saab kirjutada ühe koodi ja Arduino käiab seda ühte ja sama koodi. Samas Arduino on mõõtmetelt väga pisike ning odav. Lisaks on Arduinol olemas analoogsisendid, mida on tarvis, et mõõta vooluringis pinget.

Raspberry PI on rahakoti suurune arvuti, millel on võrreldes Arduinoga tunduvalt rohkem võimalusi, samas hind jääb võrdlemisi samaks. Raspberry PI-le on võimalik installeerida operatsioonisüsteem ning seda saab kasutada kui tavalist arvutit, samas on ta võimeline vastu võtma ja edastama signaale. Küll aga ei loe Raspberry PI analoogsignaalne, mille tõttu on Raspberry kasutamisel tarvis analoog- digitaalsignaalmuundurit tarvitada.

Raspberryt võib kasutada põhimõttelist ükskõik millise väikese projekti tuumana ning käesolevasse töösse on ta valitud oma hinna, suuruse ja lihtsuse pärast. Raspberry on vägagi algajasõbralik ning internetiavarus on täis erinevaid õpetusi, mis aitavad Raspberryt oma projekti lisada.

Töös kasutatakse Raspberry PI 4 mudel B-d, mis on kõige uuem ja võimsam Raspberry hetkeseisuga. Tegemist on seadmega, millel on kaks USB 2 ning USB 3 ühendust, kaks micro HDMI ühendust ning USB tüüp C liides, mis on mõeldud seadme toitmiseks vooluvõrgust. Raspberry PI 4 hind on umbes 60 eurot. [3] Lisaks on võimalik talle osta erinevaid lisasid, näiteks kaamera ühendus või puutetundlik ekraan.

Käesoleva projekti on valitud Raspberry, sest võrreldes Arduino mikrokontrolleriga on Raspberry-l rohkem võimalusi nii tarkvaraliselt kui ka riistvara poolel. Nimelt Raspberryle on võimalik kirjutada mitmeid erinevaid koode, mis samaaegselt toimivad ja tema

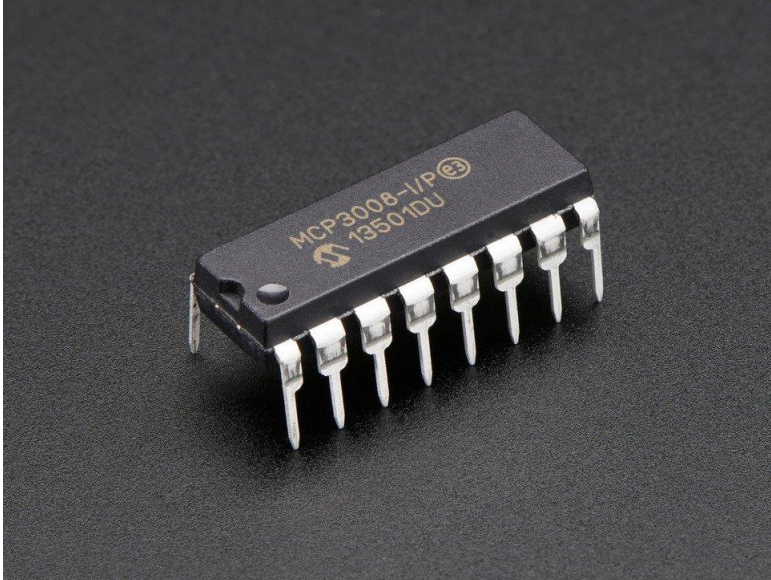
A high-angle, close-up photograph of a Raspberry Pi 4 Model B single-board computer. The green printed circuit board (PCB) is populated with various components. A large, square, silver heat sink is mounted over the central processing unit (CPU). To the left of the heat sink, the Raspberry Pi logo is visible. Below the logo, the text "Raspberry Pi" and "Model B" are printed. The board features a variety of ports: a USB-C port, a USB-A port, an HDMI port, a 3.5mm audio jack, and a 40-pin GPIO header. A black Ethernet port is located on the right side. The board is set against a plain white background.

Tegemist on 8 kanalise muunduriga, mis tähendab, et ühe seadmega on võimalik mõõta

kuni kaheksat erinevat pinget. Muundurite eesmärk on vastu võtta analoogsignaali ning muundada see Raspberryle loetavaks ühtede ja nullide jadaks ehk digitaalseks signaaliks.

MCP 3008 on 10 bittine signaaluundur ehk andmed edastatakse väärtustega 2^{10} , see tähendab, et informatsiooni vastuvõtvasse seadmesse jõuab null pinge väärtusel 0 ning

MCP 3008 konverterid on antud töösse valitud, sest tegu on üsna algajasõbralike seadeldistega, mille ühendamine lugemine on võrdlemisi lihtne ning mitmed veebiallikad soovitasid just käesolevat muundurit kasutada.



Joonis 2.3. Signaalmuundur MCP 3008 [6]

2.4. Takistid

Antud töös uuritakse, kuidas mõjutab erinevate takistite sisse ja välja lülitamine vooluringi. Kuid mis üldse on takisti ja milleks teda elektroonikas kasutatakse?

Vooluringi võib võrrelda kui vedeliku voolamist torustikus, kus pinge on veesurve, voolutugevus on toru läbimõõt ning torus olevad takistused või kitsamad kohad on takistid. Need kitsamad kohad raskendavad ja vähendavad vee liikumist. Täpselt sama kehtib ka takistite kohta – nende eesmärk on vooluringis vähendada voolutugevust kulgevas vooluringis. [7]

Takisti põhisuurust mõõdetakse oomides (Ω), mis on füüsiku ja matemaatiku Georg Ohmi auks. Kui takistus on üks oom, tähendab see, et antud takistit läbib vool üks amper, kui see on ühendatud ühe voldise pingeallikaga. Sellest tuleneb ka oomiseadus - $R = \frac{U}{I}$

Takistite puhul tuleb jälgida, et vooluringi oleks valitud õige takistusega takistid, vastasel juhul võib juhtuda, et takisti põleb ära. Igal takistil on kindel hulk soojust, mida ta suudab ära anda. Selle hulga ületamisel võibki juhtuda, et vooluring katkeb, sest takisti põleb ära. Antud soojushulka saab võtta kui võimsust, mida arvutatakse valemiga $P = U \cdot I$.

Käesolevas töös kasutatakse üheksat samasugust takistit, millede takistuseks on paberi järgi 100 oomi. Tegelikuses ei pruugi takistus kunagi 100% tõene olla, mille tõttu lisatakse takistile täpsusnäitaja. Projektis kasutatavad takistid on tolerantsiga $\pm 5\%$, mis tähendab, et reaalne takistus on vahemikus 95-105 oomi.

2.5. Lülitid

Lülitid on laialt kasutatud elektroonikakomponendid, mis annavad meile võimaluse reguleerida, kust ja millal vooluringis vool liigub. Lüliteid kasutatakse peaaegu igas vooluringis, kus on vaja elektri ahelas mingid osad katkestada või ühendada. Näiteks uksekella vajutades, aktiveerib lüliti, mis lubab voolul liikuda edasi seadeldiseni, mis tekitab heli. [8]

Lüliteid eksisteerib kahte sorti – fikseeruvad ja mittefikseeruvad. Fikseeruvad lülitid on sellised, kus lüliti aktiveerimisel muudab lüliti olekut ning see olek jääb püsima. Mitte fikseeruva lüliti puhul olek on muudetud ainult sellel hetkel, kui lüliti vajutatakse. [8]

Lülitid saavad olla kas *normally-off* või *normally-on*. Esimesel juhul ei ole algolekus ühendust ehk vool ei läbi lüliti. *Normally-on* lüliti puhul on ühendus olemas, kuid lüliti aktiveerimisel ühendus katkeb. [8]

Käesolevas töös kasutatakse tumbler *on-off* lüliti, mis on mõeldud kasutamiseks kuni 1,5 amperi suuruse vooluga, ning maksimaalne voolupinge võib olla 250 V. Antud lüliti on valitud oma suuruse ja hinna tõttu. Kuna seadeldist üritatakse hoida võimalikult pisikesena, siis antud lülitid sobivad hästi.



Joonis 2.4. Projektis kasutatavad lülitid [9]

Lülitite eesmärk antud vooluringis on reguleerida, millistest harudes on võimalik voolul liikuda. See tähendab, et lülitite sisse välja lülitamisega aktiveeritakse erinevaid takisteid, ning see annab võimaluse kasutajale seadmega suhelda.

3. SEADME VALMISTAMINE

3.1. Raspberry PI ülesseadmine

Raspberry PI ei ole arvuti, mille saab ostes seina lükata ja kasutama hakata, sest Raspberry l puudub kõvaketas. See tähendab, et Raspberry on vaja enne kasutamist ülesseadistada. Talle on vaja paigaldada operatsioonisüsteem ning teha seal vajalikud toimetused, et Raspberry oleks kasutusvalmis

3.1.1. Raspberry PI esmased sätted

Raspberry PI kasutab mäluna *Micro SD* kaarti, kuhu on vaja installeerida operatsioonisüsteem. Käesolevas töös kasutatakse 16 gigabaidist *Micro SD* kaarti, kiirusklassiga 10, mis tähendab, et maksimaalne kaardi kirjutuskiirus on 10 megabaiti sekundis. [10]

Esmalt on vaja *SD* kaardile paigaldada operatsioonisüsteem. Käesolevas projektis kasutatakse *Raspberry Pi OS (32-bit)* operatsioonisüsteemi, mis põhineb avatud lähtekoodiga Debian operatsioonisüsteemil.

Raspberry PI OS süsteemi paigaldamiseks kasutatakse eraldiseisvat lauaarvutit, mille külge tuleb ühendada *Micro SD* kaart. Selle jaoks kasutab autor USB pesasse käivat adapterit, mis tuli kaasa *Micro SD* kaardi ostuga.

Operatsioonisüsteemi installeerimiseks *SD* kaardile kasutatakse Raspberry kodulehelt allalaaditud programmi *Raspberry PI Imager*, mis teeb protsessi ülimalt lihtsaks: on vaja valida soovitud operatsioonisüsteem, näidata programmile, millisele andmetalletusseadmele see lahti pakkida ning ülejäänud teeb programm ise.

Kui programm on lõpetanud operatsioonisüsteemi paigaldamise on vaja see paigaldada Raspberry l asuvasse kaardipesasse ning Raspberry ühendada vooluvõrku. Lisaks ühendatakse USB pesadesse arvutihiir kui ka klaviatuur, tänu millele saab seadeldist juhtida.

Esmasel käivitamisel küsib Raspberry millisest riigist ja ajatsoonist kasutaja on, et oleks paigas kellaaeg ning Raspberry keel. Lisaks küsitakse millist klaviatuuriseadistust kasutada ning palutakse seadmele salasõna genereerida.

Raspberry PI on võimalik ühendada ka *WIFI* internetiühendusega või *Ethernet* kaabliga. See tähendab, et Raspberry l edasisi toiminguid tehes on võimalik vaadata õpetusi otse Raspberry l t ning ei pea kasutama selle jaoks eraldi arvutit.

3.1.2. SPI liidese lubamine

Raspberry Pi kasutab erinevate seadmetega suhtlemiseks SPI suhtlusprotokolli ehk *Serial Peripheral Interface*. Näiteks analoog- digitaalmuunduriga suhtlemiseks on vaja samuti kasutada SPI liidest. SPI kasutab suhtlemiseks nelja eraldi ühendust ehk klemmi, mis on järgnevad:

1. CLK (*Serial clock*);
2. MISO (*Master input slave output*);
3. MOSI (*Master Output slave input*);
4. CS (*Chip select*). [11]

CLK ühenduse puhul toimub signaalide pulseerimine kindlal sagedusel. Antud sagedusel vahetavad Raspberry Pi ja tema külge ühendatud seadeldis infot. [11]

MISO ühendust kasutab Raspberry PI, et lugeda talle saadetud informatsiooni, kus informatsiooni saatjaks on tema külge ühendatud seadeldis. Info vahetus toimub samal sagedusel, millel pulseerib CLK signaal. [11]

MOSI klemmi kasutab Raspberry PI, et edastada informatsiooni. See tähendab, et antud klemmiga on võimalik öelda tema külge ühendatud seadmele, mida teha. [11]

CS klemm ütleb, millist seadeldist kasutada. Kuna SPI klemmidele võib ühendada samaaegselt mitu erinevat seadeldist, siis CS klemm ütleb arvutile, millise seadeldisega parasjagu kontakt toimub ning sel juhul suhtlevad teised SPI ühendused just selle kindlaks määratud seadmega. [11]

Raspberry Pi-l ei ole SPI klemmide kasutamine algupäraselt lubatud, vaid need on vaja seadetest sisse lülitada. Selle jaoks on vaja käsuterminal avada ning kirjutada sinna järgnev koodirida:

```
sudo raspi-config [12]
```

Koodi sisestamisel avaneb menüü, kus saab Raspberry-t konfigureerida. Valikus on üheksa erinevat punkti, millest üks on *interface options*. Vajutates klaviatuuril *Enter* klahvi, avaneb järgmine menüü, kus on erinevate liideste sisse ja välja lülitamise valikud. Antud menüüs tuleb valida SPI ning *Enable*. Viimaseks on vaja arvuti taaskäivitada, et muudatused saaksid aset võtta. [12]

3.2. Analoo- digitaalmuunduri ühendamine

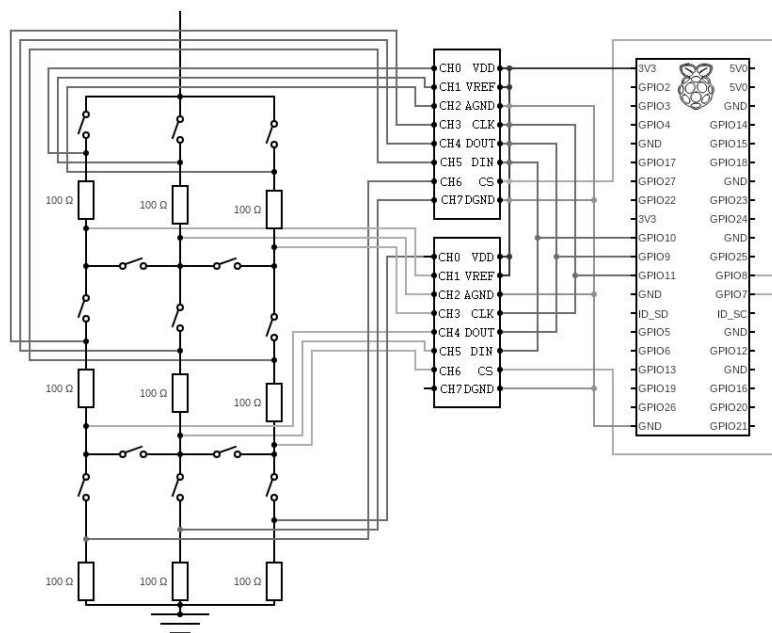
3.2.1. MCP 3008 füüsiline ühendamine

Raspberry ja MCP 3008 omavahelise suhtluse saavutamiseks tuleb seadeldised ühendada. Selle jaoks on Raspberry varustatud *GPIO*-ga ehk mitmeotstarbeline sisendite ja väljundite moodul. Moodul koosneb neljakümnest erinevast klemmist, mida kasutajal on võimalik kasutada kas sisendi- või väljundina. Mõned kindlad klemmid on määratud kas 3.3 või 5 voldise väljundpingega või siis maandusena.

MCP 3008 on varustatud 16-ne klemmiga, millest kaheksa on mõeldud ühendamiseks mõne teise seadmega, näiteks Raspberry. Need klemmid tuleb korrektselt ühendada kindlate Raspberry PI väljund-ja sisendklemmidega ning siis on võimalik Raspberryle edastada vastavaid signaale.

Analoogdigitaalmuunduri klemmid number 1 kuni 8 on mõeldud analoogsignaali mõõtmiseks. See tähendab, et nendele klemmidele tuleb ühendada voluringi need osad, millede pingeid soovitakse mõõta. Klemmid 9 – 16 on mõeldud suhtlemiseks teiste seadmetega. [8]

Seadmete omavaheliseks ühendamiseks kasutatakse elektriskeemi, mis on autori poolt loodud vabavaralise veebipõhise programmiga *Circuit-Diagram*. Antud tarkvaraga on võimalik luua lihtsasti elektriskeeme, lisaks on paljud seadmed juba programmis sees, mis tähendab, et kulub vähe aega seadmete välja joonestamisele.



Joonis 3.1. Seadmete ühendusskeem

Tabel 3.1 näitab, millised Raspberry PI ühenduspinnid tuleb kokku viia signaalmuunduri pinnidega.

Tabel 3.1. Kuidas ühendada signaalmuundur Raspberry-ga [13]

Raspberry Pi klemmid	MCP 3008 Klemmid
Klemm 17, 3.3 V toide	VDD
Klemm 17, 3.3 V toide	V _{REF}
Klemm 14, maandus	AGND
Klemm 23, GPIO 11	CLK
Klemm 21, GPIO 9	D _{OUT}
Klemm 19, GPIO 10	D _{IN}
Klemm 24, GPIO 8	CS/SHDN
Klemm 14, maandus	DGND

3.2.2. Analoo- digitaalmuunduri installeerimine Raspberry-s

Ei piisa ainult signaalmuunduri ja Raspberry omavahelisest füüsilisest ühendusest. Signaalmuundurilt informatsiooni kätte saamiseks on vaja Raspberry PI-le selgeks teha, kuidas ja millistelt klemmidelt infot lugeda. Selle jaoks tuleb installeerida arvutisse lisapakett, mis õpetab arvuti signaalmuunduriga kommunikeerima ning programmeerida kood, kus öeldakse arvutile millistelt kanalitelt ta informatsiooni lugema peab ja mida nende andmetega edasi teha.

Töö koostamisel kasutatakse *Python 3* programmeerimiskeelt. Selle jaoks on vaja paigaldada seadmesse tarkvara nimega *pip3*. Tegemist on tarkvaraga, mis võimaldab alla laadida ja konfigureerida *Python*-i mooduleid. [14] *Pip3* installeerimiseks tuleb juhtterminal avada ning sisestada sinna järgnev kood:

```
sudo apt-get install python3-pip [15]
```

Kui tarkvara on lõpetanud allalaadimise ja installeerimise on võimalik seadmesse paigaldada *adafruit-blinka*. Selle jaoks tuleb juhtterminali kirjutada kood

```
sudo pip3 install adafruit-blinka [15]
```

Nüüd kasutas arvuti *pip3* tarkvara, sest *adafruit-blinka* on *pythonile* lisatav moodul. Viimaseks installeeritakse MCP 3008 tarkvara, et arvutil oleks võimalik suhelda signaalimuunduriga. Juhtterminali on vaja sisestada edasisene:

```
sudo pip3 install adafruit-circuitpython-mcp3xxx [15]
```

Kui kõik kolm etappi on korrektselt täidetud saab kirjutama hakata juhtkoodi, mida arvuti mõistab korrektselt ning tänu millele on ta võimeline signaalimuundurilt korrektseid andmeid kätte saama. [15]

Juhtkoodi kirjutamise osas, kus Raspberry PI loob ühenduse analoog-digitaalsingaari muuturiga, kasutatakse suuresti *Raspberrypi-spy.co.uk* kasutaja, Matt, koodi. [16]

3.3. 3D prinditava detaili loomine

Raspberry PI ja voluring paigutatakse 3D prinditud karbi sisse, kus on piisavalt ruumi Raspberlyle, kui ka juhtmetele, lülititele ning analoog-digitaalkonverteritele. Lisaks hoiab karp tervet süsteemi turvaliselt kinni. 3D mudeli loomiseks kasutatakse Autodesk Fusion 360 programmi, millega autor on varasemalt põgusalt kokku puutunud.

3D prinditav hoidiku mõõtmed on 124 mm korda 120 mm, Seinapaksuseks valitakse 2 mm ehk reaalselt põhjapinda on 120 mm korda 116 mm. Selliste dimensioonide puhul on piisavalt ruumi, et hoidiku põhja paigutada Raspberry PI ning lisaks monteerimisplaat, kuhu saab turvaliselt ühendada Raspberry ja MCP 3008 signaalimuunduri.

Karbi põhjas peab olema neli kruvimi auku, kuhu külge saab Raspberry kinnitada. Monteerimislauda saab kinnitada näiteks liimiga või samuti kruvidega. Karbi kõrguseks on 52 mm, 2 mm on hoidiku põhi, niiet 50 mm jääb ruumi seadmete ja juhtmete monteerimiseks.

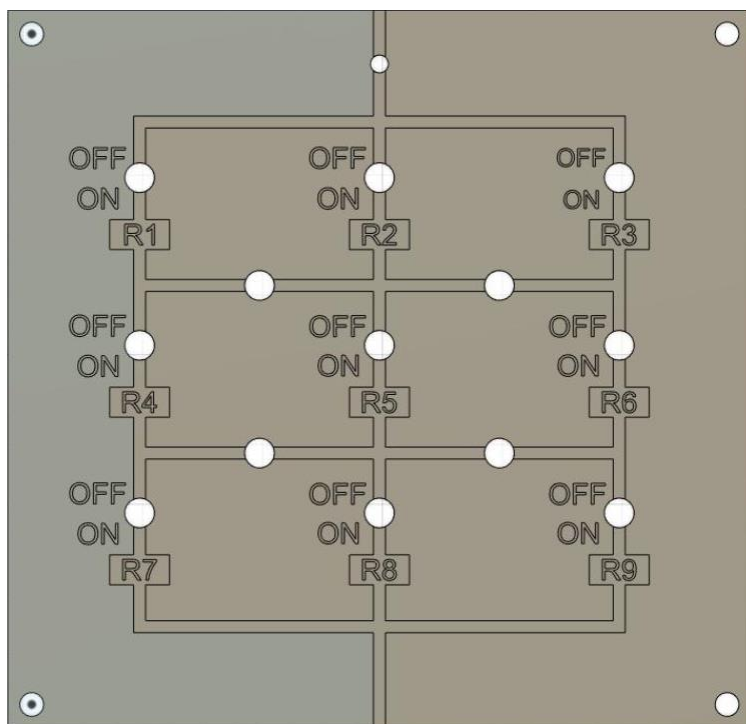
Karbi külgedele tuleb õigetes kohtadesse modelleerida avad, kust kaudu saab Raspberry ühendada elektrivõrku ning ekraani külge. Igaks juhuks kujundatakse karbile ka USB ühenduste avad, millega saab Raspberry ühendada hiire ja klaviatuuriga, et vajadusel teha programmis muudatusi.

Kõige keerulisem on karbi kaane loomine, sest sinna peavad olema ühendatud lülitid ning kaanele kujundatakse reljeef, mis näitab voolu liikumise suunda. See tähendab, et on vaja modelleerida elektriskeem karbi peale, kus õigetes kohtades on lülitite avad, takistid ning kujutatavad juhtmed.

3D mudeli valmimisel salvestatakse see *.stl* formaati, mida 3D printer lugeda oskab. Printimiseks kasutatakse 3D printerit Creality Ender 3 pro.



Joonis 3.2. Karbi põhi Fusion 360-s



Joonis 3.3. Karbi ülemine osa Fusion 360-s

4. TARKVARALISE KASUTAJALIIDESE LOOMINE

4.1. *Python* programmeerimiskeel

Python on 1991. Aastal loodud programmeerimiskeel Guido Van Rossumi poolt Hollandis. Alates sellest ajast on *Python* pidevalt saanud uuendusi ning välja on antud erinevaid versioone. 2001. aastal loodi mittetulundusühing *Python software foundation*, mis loodi eesmärgil, et omada *Pythoniga* seotud intellektuaalseid omandeid. [16]

Pythoni lähtekood on avatud, mis tähendab, et kõik võivad seda muuta ja kasutada, kuidas ise soovivad, mistap on loodud *Pythonile* väga mitmeid mooduleid ja lisasid hoopis teiste programmeerijate poolt. [16]

Raspberry PI-st kasutatakse koodi kirjutamiseks tarkvara *Thonny Python IDE*. Antud tarkvara tuleb kaasa operatsioonisüsteemiga, mis käesolevas projektis Raspberry-le on paigaldatud. Tegu on algajasõbraliku, kuid siiski võimeka programmiga.

4.2. Tkinter

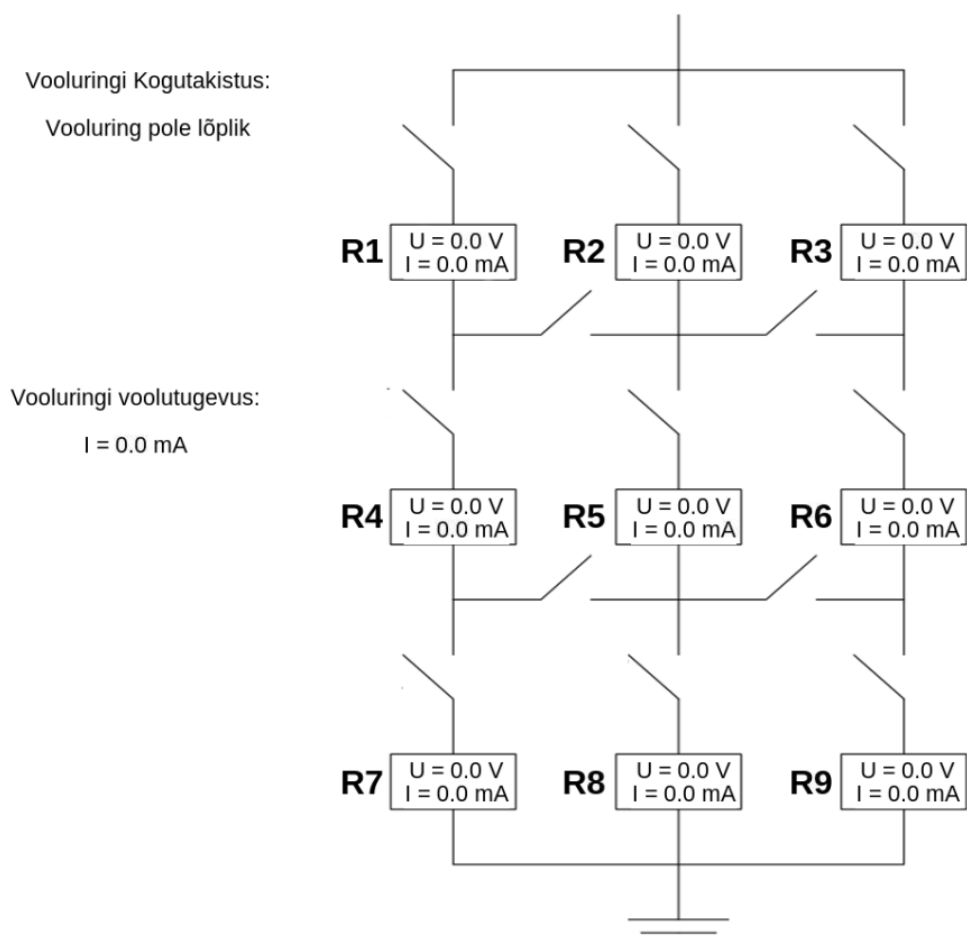
Pythonis on vaja kasutajaliidese loomiseks abi erinevatelt *GUI* ehk *graphical user interface* moodulitelt. Nimelt *Pythonile* on loodud mitmeid vahendeid, millega kasutajaliidest luua. Nendest populaarsemad on loetletud järgnevalt:

1. *PyQt5*;
2. *Tkinter*;
3. *Kivy*;
4. *wxPython*;
5. *Libavg*;
6. *PySimpleGUI*. [17]

Antud projekti tegemisel proovis autor kasutajaliidese loomiseks mitmeid erinevaid mooduleid, näiteks *PySimpleGUI*, *PyQt5* ning *Tkinter*. Lõpuks otsustas autor jääda *Tkinter*it kasutama, sest selle kasutamine tundus kõige lihtsam ning olemas on kõik võimalused, et luua lihtne kasutajaliides, mis oleks arusaadav kasutajale.

Kasutajaliidese eesmärk antud projektis on kasutajale näidata, milliseid takisteid läbib vool ning kui suur on voolutugevus ja pinge takistitel. Kuvatud peab olema ka vooluringi kogutakistus.

Programmi teeb lihtsaks asjaolu, et ta peab ainult informatsiooni ekraanil kuvama, sest tarkvara poolel ei toimu mingit kasutaja sisendit. See tähendab, et ei ole vaja luua keerulisi funktsioone, kus kasutaja sisendi tulemusel peab arvuti midagi tegema.



Joonis 4.1. Kuvatõmmis kasutajaliigest ilma vooluringi ühenduseta.

4.3. Ekraanil kuvatavate väärtuste leidmine

4.3.1. Pinge mõõtmine

MCP 3008 on 10 bittine analoog-digitaalmuundur, mis tähendab, et see annab annab mõõdetud tulemused täpsusega 2 astmel 10 ehk 1024. See tähendab, et Raspberrysse jõuavad mõõdetud väärtused vahemikus 0 – 1023, kus 0 tähendab, et pinge on 0 volti ning 1023 tähendab 3.3 volti.

Pinge teisendamiseks on vaja koodi lisada osa, kus arvuti teisendab mõõdetud tulemuse korrektseks pingeväärtuseks.

$$U = \frac{U_m}{1023} \cdot 3.3,$$

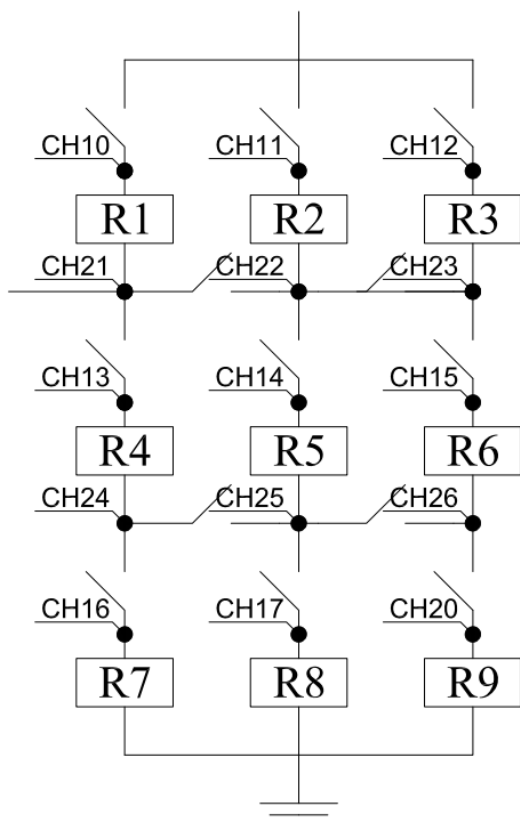
Kus,

U on korrektne pinge väärtus;

U_m on mõõdetud väärtus vahemiks 0 kuni 1023.

Edasi tuleb arvestada asjaolu, et seadeldis ei mõõda pingelangusid takistitel, vaid pinget, mis antud punktis on suhtes maandusega. See tähendab, et ainult takistid R_7 , R_8 ning R_9 näitavad reaalselt pingelangu, sest nende takistite teine osa on ühenduses maandusega.

Ülejäänud takistite puhul tuleb arvutada pingelang, mille leidmine on võrdlemisi lihtne, sest pinge mõõdetakse takisti mõlemas otsas. Skeem, kuidas pingeid mõõdetakse on järgnev:



Joonis 4.2. Analoo- digitaalmuunduri mõõteskeem

Joonisel on välja toodud, millistest punktidest pingeid mõõdetakse ning kuhu on antud punkt ühendatud. Näiteks CH10 tähendab, et kasutatakse esimese signaalmuunduri kanal nulli. CH21 tähendab, et kasutatakse teise signaalmuunduri esimest kanalit.

Takisti R1 pingelangu leidmiseks on vaja lahutada CH10-st CH21 väärtus. Edasi tuleb antud väärtus teisendada korrektseks pingeväärtuseks ning kuvada ekraanil korrektses kohas.

4.3.2. Voolu tugevuse ja kogutakistuse arvutamine

Üks töö eesmärkidest on kuvada ekraanil voolutugevus, mis igat takistit läbib, et näidata kuidas Kirchhoffi seadused töötavad. On teada, et takistite väärtus on 100 oomi ning

pingelang igal takistil on juba korrektselt välja arvutatud. Edasi tuleb kasutada oomiseadust : takisti pingest jagada takisti väärtus ehk 100 oomi.

Antud töös on voolud niivõrd väikesed, kui toitepingeks on 3.3 volti ning kogutakistus 100 oomi, siis voolutugevus tuleks 0.033 A, mistap esitatakse andmed milliamprites ehk sajaga jagamise asemel korrutatakse 10-ga. Nii tuleks hoopiski vastus 33 mA.

Kogutakistuse leidmiseks on vaja kasutada jällegi oomiseadust. Vooluringi kohta on juba teada kogupinge, pingelangud ning voolutugevused iga tarbija juures. Kogutakistuse leidmiseks on vaja kasutada kogupinget ning vooluringi voolutugevust. Voolutugevuse leidmiseks on vaja ühe rea takistite voolutugevused kokku liita ehk näiteks $I_1 + I_2 + I_3 = I$

Edasi on teada, et pinge toiteallikast on 3.3 volti, mis tähendab, et kasutades oomiseadust on võimalik leida kogutakistus:

$$R = \frac{U}{I} = \frac{3.3}{I},$$

Kus

R - vooluringi kogutakistus;

U - toiteallika pinge, mis on konstantne 3.3 volti;

I – voolutugevus.

5. SEADME MONTEERIMINE JA KASUTAMINE

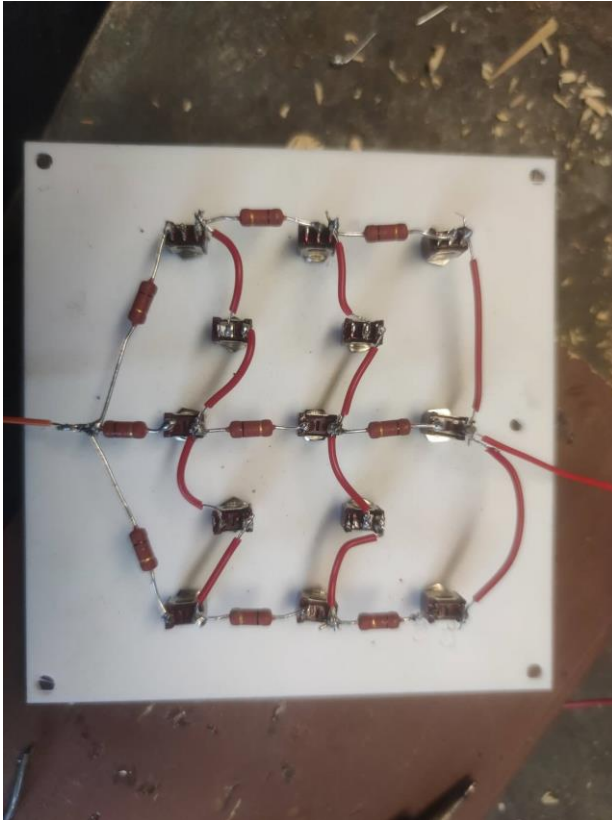
5.1. Seadme kokkupanek

Viimaseks tuleb seadeldis kokku panna. See tähendab, et Raspberry PI tuleb kinnitada karbi põhja. Juhtmete ühendamise lihtsustamiseks paigaldatakse karpi ka maketeerimislaud, tänu millele on analoog- digitaalmuunduri ühendamine Raspberry PI külge märkimisväärselt lihtsam.



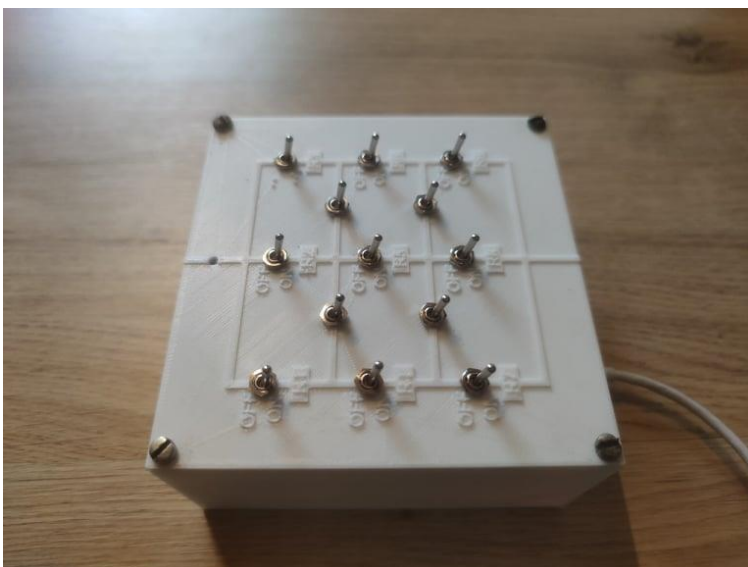
Joonis 5.1. Karbi põhi koos sinna paigaldatud Raspberry PI ning maketeerimislauga

Edasi on vaja lülitid kruvida karbi kaane külge ning joota lülitite külge takistid, et tekiks korrektne vooluring. Juhtmed joodetakse jootekolbiga, kasutades jootetina ning jooteainet. Lisaks tuleb joota analoog- digitaalmuunduri mõõtekanalite juhtmed takistite külge ning ühendada need maketeerimislaual, et oleks võimalik takistite pingelangusid mõõta.



Joonis 5.2. Takistite jootmine lülitite külge

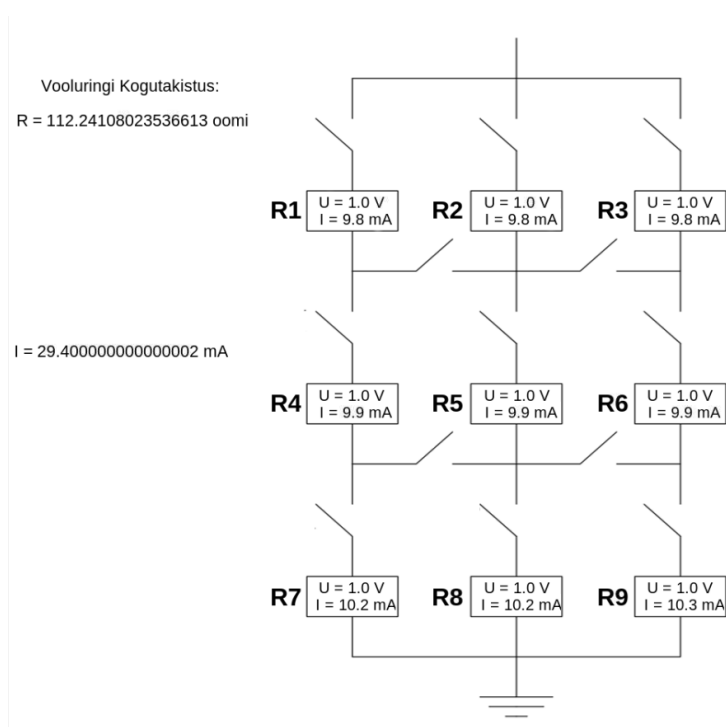
Peale jootmist kinnitatakse nelja kruviga karbi kaas põhja külge ning testitakse, kas seadeldis toimib. Karbi kinnitamisel selgus, et kaks juhet polnud korralikult vooluringikülge joodetud, vaid tulid lahti, mille tõttu oli vaja need juhtmed uuesti joota.



Joonis 5.3. Valmis produkt

5.2. Seadme kasutamine

Viimaseks on võimalik seadet testida, et näha, kas ekraanil kuvatakse kõik väärtused korrektselt. Selle jaoks lülitab töö autor kõik lülitid *on* asendisse ning käivitab Raspberrys kirjutatud programmi. Saadud pilt on joonisel 5.4.

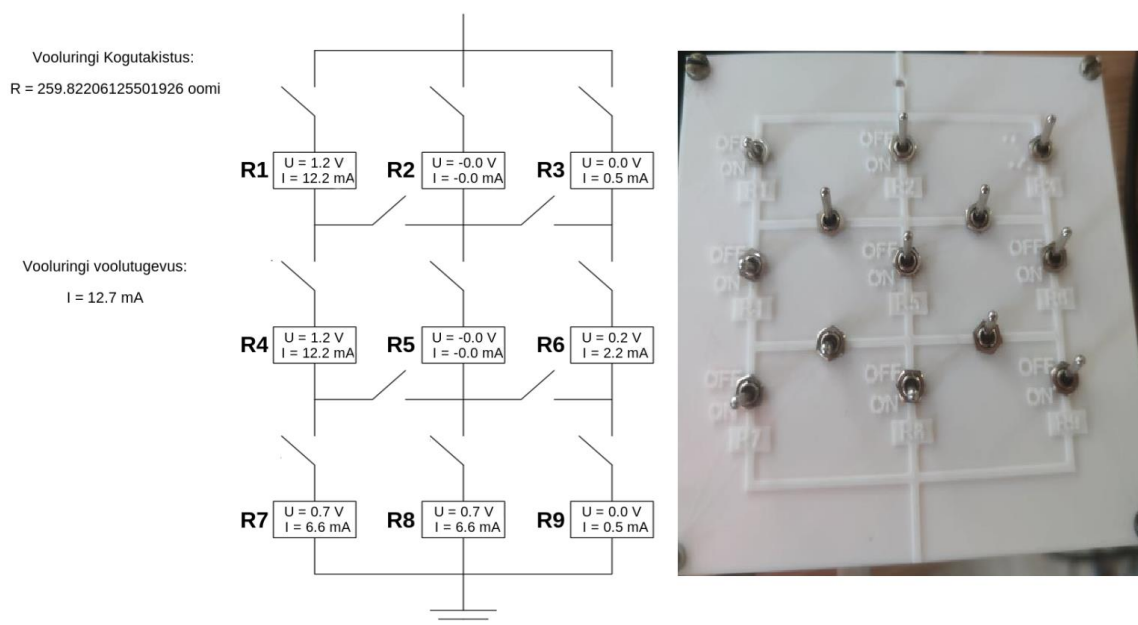


Joonis 5.4. kuvatõmmis ekraanil olevatest väärtustest, kui kõik lülitid *on* asendis

Jooniselt on näha, et kõikide takistite pingeväärtuseks on 1.0 volti. See ei ole küll päris täpne tulemus, kuvatav väärtus peaks olema 1.1 volti. Takistite voolutugevuseks on keskmiselt 10 mA, mis pole samuti päris täpne tulemus, peaks olema 11 mA. Kuvatud kogutakistus on 112.2 oomi. Arvutuste kohaselt peaks olema tulemuseks 100 oomi ning vooluringi voolutugevuseks on seadeldis arvutanud 29.4 mA.

Töö autor ei pea saadud tulemusi probleemiks, sest tuleb arvestada asjaoluga, et iga takisti ei ole täpselt 100 oomi ning analoog- digitaalmuunduril on samuti mingit sorti mõõtemääramatus.

Järgmise testi teeb autor olukorras, kus vool läbib takistit R1, R4, R7 ning R8. Tulemused koos pildiga lülititest on joonisel 5.5.



Joonis 5.5. kuvatõmmis ekraanil olevatest väärtustest koos vastavate lülite paigaldusega

Antud mõõtmistega hakkas silma asjaolu, et programmi arvates on takistil R6 pingeks 0.2 volti ning voolutugevuseks 2.2 mA. See on väär tulemus, sest takisti R6 on praegusel juhul vooluringist välja lülitatud.

Lisaks märkas autor, et mõned takistid, mis pole hetkel vooluringis, näitavad vahepeal tulemusi kuni 0.5 volti. See võib olla tingitud asjaolust, et analoog- digitaalmuundur on liiga tundlik. Teine võimalus on, et juotmisel on tekkinud probleemseid kohti, kus erinevate juhtmete elektrit juhtiv osa puutub kokku teise juhtmega.

Üldjuhul näitab programm õigeid tulemusi, millega võib rahule jääda, sest tema funktsioon on täidetud. Jooniselt on selgesti näha parameetrid vooluringi kohta ning välja tulevad ka seaduspärasused, näiteks voolutugevus takistil R4 on 12.2 mA ning takistitel R7 ja R8 6.6 mA ehk peaaegu täpselt pool.

Programmis oleks vaja teha ka väiksemaid muudatusi, näiteks kogutakistus ja vooluringi voolutugevus võiksid olla kuvatud ühe koma kohaga ja lisada tasuks ka informatsioon, et vooluringi toitepinge on 3.3 volti.

KOKKUVÕTE

Käesoleva bakalaureusetöö eesmärgiks oli koostada seadeldis, millega on võimalik uurida vooluringi seaduspärasusi, samal ajal kui kasutajal on võimalik vooluringi olemust pidevalt muuta. Selle jaoks koostati vooluring, mida oleks võimalik lülitite sisse-välja lülitamisega juhtida, valiti komponendid, mis suudaksid vooluringist informatsiooni vastu võtta ning tulemust ekraanil kuvada.

Seadme tuumaks ehk arvutiks valiti Raspberry PI. Edasi seadistati Raspberry Pi kasutustuskõlblikuks ning lisati talle külge, ja seadistati töövalmis, kaks MCP 3008 analoog- digitaalmuundurit, sest Raspberry PI-l puuduvad analoogsisendid pingete lugemiseks. Pidades silmas Raspberry PI suurust ning ühendamistele kuluvat ruumi loodi 3D printerile mudel hoidikust, kuhu sisse oleks võimalik komponendid paigutada.

Järgnevaks koostati tarkvaraline kasutajaliides, mille eesmärk on kasutajale anda pidevalt informatsiooni vooluringis toimuva kohta. See tähendab, et ekraanile kuvati iga takisti pinge ja voolutugevus, lisaks kogutakistus ja vooluringi voolutugevus.

Viimaseks monteeriti seadeldis valmis, mille jaoks ühendati analoog- digitaalmuundurid karbis raspberry PI-ga. Töö lihtsuse mõttes kasutati maketeerimisplaati, mis oli samuti paigutatud karbi põhja. Viimaseks paigaldati lülitid kaane sisse ning ühedati analoog- digitaalmuundurile juhtmed, mis joodeti vooluringi iga takisti külge, et mõõta nende pingelangusid ning katsetati, kas programm näitab korrektseid tulemusi.

KASUTATUD KIRJANDUS

1. **Bhatia, A.** Fundamentals of Direct Current Circuits. [veebiallikas]
<https://www.cedengineering.com/userfiles/Fundamentals%20of%20DC%20Circuits.pdf>
 (21.05.2021)
2. **Mahmoud, M.** (2018). Advanced Control Design with application to Electromechanical Systems. Butterworth-Heinemann (2018.) 388 lk. [veebiraamat]
<https://www.sciencedirect.com/topics/engineering/kirchhoff-law> (21.05.2021)
3. Raspberry PI 4 Specifications. [kodulehekülg]
<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/> (17.05.2021)
4. Raspberry PI 4 B 4 GB 4 x 1.5 GHZ Raspberry PI. [veebipood]
<https://www.conrad.com/p/raspberry-pi-4-b-4-gb-4-x-15-ghz-raspberry-pi-2138865>
 (17.05.2021)
5. MCP3004/3008. Kasutusjuhend. [veebiallikas] <https://cdn-shop.adafruit.com/datasheets/MCP3008.pdf> (10.05.2021)
6. Adafruit MCP3008 – 8 – Channel 10- Bit ADC With SPI Interface. [veebipood]
<https://www.adafruit.com/product/856> (10.05.2021)
7. **Sinivee, V.** (2004). *Elektroonika-aabits* [veebiallikas]
<http://parsek.yf.ttu.ee/~felc/ak/Aabits1.pdf> (08.05.2021)
8. **Papiewski, J.** (2018). How Do Push Button Switches Work in an Electrical Circuit? [veebiallikas] <https://sciencing.com/push-switches-work-electrical-circuit-5030234.html>
 (16.05.2021)
9. Tumbler 2*ON-OFF-ON 1.5A 250V PCB horisontaalne [veebipood]
https://www.oomipood.ee/product/tssm203_2c3_tumbler_2_on_off_on_1_5a_250v_pcb_horisontaalne?sort=p.price&order=ASC&limit=96 (16.05.2021)
10. **SD-3C LLC.** SD Standard for Video Recording. (2021). [veebiallikas]
<https://www.sdcard.org/developers/sd-standard-overview/speed-class/> (14.05.2021)
11. **Artisan's Asylum.** Introduction to Raspberry Pi. [veebiallikas] <https://raspberrypi-aa.github.io/session3/spi.html> (14.05.2021)
12. **Matt.** Enable SPI Interface on the Raspberry Pi. (2014). [veebiallikas]
<https://www.raspberrypi-spy.co.uk/2014/08/enabling-the-spi-interface-on-the-raspberry-pi/> (14.05.2021)

13. **Matt.** Analogue Sensors On The Raspberry Pi Using An MCP3008. [veebiallikas]
<https://www.raspberrypi-spy.co.uk/2013/10/analogue-sensors-on-the-raspberry-pi-using-an-mcp3008/> (14.05.2021)
14. **MH Themes.** What is Pip3? (2021). [veebiallikas] <https://vgkits.org/blog/what-is-pip3/> (14.05.2021)
15. **Sklar, M.** Analog Inputs for Raspberry PI Using the MCP3008. (2012). [veebiallikas]
<https://learn.adafruit.com/reading-a-analog-in-and-controlling-audio-volume-with-the-raspberry-pi/necessary-packages> (14.05.2021)
16. **Python Software Foundation.** History and License. (2021). [veebiallikas]
<https://docs.python.org/3/license.html> (18.05.2021)
17. **Costa, D, C.** Top 10 Python GUI Frameworks for Developers. (2020). [veebiallikas]
<https://towardsdatascience.com/top-10-python-gui-frameworks-for-developers-adca32fbe6fc> (18.05.2021)

LISAD

Lisa 1. Programmi juhtkood

```
# Vajalike Pakettide Import
import spidev
import time
import os
from tkinter import *
import tkinter.font

# SPI Avamine, Chip Select (CS) Pinni Valimine
spi = spidev.SpiDev()
spi.open(0,0)
spi.max_speed_hz=1000000

spi2 = spidev.SpiDev()
spi2.open(0,1)
spi2.max_speed_hz=1000000

# Nähtava Akna Loomine
win = Tk()
win.title("takistiruudustik")
win.geometry("1080x1980")
myFont = tkinter.font.Font(family = 'Helvetica', size = 16,)
myFont2 = tkinter.font.Font(family = 'Helvetica', size = 24, weight = "bold")

# Taustapildi Paigaldamine
bg = PhotoImage(file="Background2.png")
Taustapilt = Label(win, image = bg)
Taustapilt.place(x=0, y=0, relwidth=1, relheight=1,)

# Funktsioon, Mis Loeb Andmeid MCP 3008-lt.
# Kasutusel Kaks Funktsiooni, Esimene Loeb CS 1-lt, Teine CS 2-lt
def ReadChannel(channel):
    adc = spi.xfer2([1,(8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

# Takisti Väärtus
takisti = 100

def ReadChannel2(channel):
    adc = spi2.xfer2([1,(8+channel)<<4,0])
    data = ((adc[1]&3) << 8) + adc[2]
    return data

# Funktsioonid, Mis Muudavad Mõõdetud Tulemuse Pingeks Ja Vooluks
def ConvertVolts(data,places):
```

Lisa 1 järg

```
volts = (data * 3.3) / float(1023)
volts = round(volts,places)
return volts
```

Takistile Antakse Väärtuseks, Mis Parasjagu Vooluringi Takistused On Ning
Korrutatakse 1000-ga, Et Saada Milliamprid

```
def ConvertAmperage(data,places):
    amperage = (data * 3.3) / float(1023) / takisti * 1000
    amperage = round(amperage,places)
    return amperage
```

Loeb Kindalt Channelilt

```
voltage_ch0 = 0
voltage_ch1 = 1
voltage_ch2 = 2
voltage_ch3 = 3
voltage_ch4 = 4
voltage_ch5 = 5
voltage_ch6 = 6
voltage_ch7 = 7
```

```
def GUIloop():
```

Tulemuste Defineerimine Eraldi Nimetustele

```
voltage_ch0_level = ReadChannel(voltage_ch0)
voltage_ch1_level = ReadChannel(voltage_ch1)
voltage_ch2_level = ReadChannel(voltage_ch2)
voltage_ch3_level = ReadChannel(voltage_ch3)
voltage_ch4_level = ReadChannel(voltage_ch4)
voltage_ch5_level = ReadChannel(voltage_ch5)
voltage_ch6_level = ReadChannel(voltage_ch6)
voltage_ch7_level = ReadChannel(voltage_ch7)
voltage_ch0_level2 = ReadChannel2(voltage_ch0)
voltage_ch1_level2 = ReadChannel2(voltage_ch1)
voltage_ch2_level2 = ReadChannel2(voltage_ch2)
voltage_ch3_level2 = ReadChannel2(voltage_ch3)
voltage_ch4_level2 = ReadChannel2(voltage_ch4)
voltage_ch5_level2 = ReadChannel2(voltage_ch5)
voltage_ch6_level2 = ReadChannel2(voltage_ch6)
```

Korrektse Pingelangu Arvutamine

```
voltage0_pärisväärtus = ReadChannel(voltage_ch0) - ReadChannel2(voltage_ch1)
voltage1_pärisväärtus = ReadChannel(voltage_ch1) - ReadChannel2(voltage_ch2)
voltage2_pärisväärtus = ReadChannel(voltage_ch2) - ReadChannel2(voltage_ch3)
voltage3_pärisväärtus = ReadChannel(voltage_ch3) - ReadChannel2(voltage_ch4)
voltage4_pärisväärtus = ReadChannel(voltage_ch4) - ReadChannel2(voltage_ch5)
voltage5_pärisväärtus = ReadChannel(voltage_ch5) - ReadChannel2(voltage_ch6)
```

Lisa 1 järg

```
# Voolutugevus Viimase Kolme Takisti Juures
```

```
Vooltakistil1 = ConvertAmperage(voltage0_pärisväärtus, 1)
```

```
Vooltakistil2 = ConvertAmperage(voltage1_pärisväärtus, 1)
```

```
Vooltakistil3 = ConvertAmperage(voltage2_pärisväärtus, 1)
```

```
# Vooluringi Voolutugevus
```

```
voolutugevus = ((Vooltakistil1) + Vooltakistil2 + Vooltakistil3)
```

```
# Vooluringi Kogutakistus
```

```
kogutakistus = 3.3 / (voolutugevus + 0.001) * 1000
```

```
#Andmete Ning Teksti Paigutamine Ekraanile
```

```
R1 = Label(win, text = "R1", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R1.place(x=340, y=267)
```

```
R2 = Label(win, text = "R2", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R2.place(x=550, y=267)
```

```
R3 = Label(win, text = "R3", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R3.place(x=765, y=267)
```

```
R4 = Label(win, text = "R4", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R4.place(x=340, y=518)
```

```
R5 = Label(win, text = "R5", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R5.place(x=550, y=518)
```

```
R6 = Label(win, text = "R6", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R6.place(x=765, y=518)
```

```
R7 = Label(win, text = "R7", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R7.place(x=340, y=770)
```

```
R8 = Label(win, text = "R8", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R8.place(x=550, y=770)
```

```
R9 = Label(win, text = "R9", font = myFont2, bg = 'white', height = 1, width = 2)
```

```
R9.place(x=765, y=770)
```

```
Takistipinge1 = Label(win, text = "U = " + format(ConvertVolts(voltage0_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

```
Takistipinge1.place(x=400,y=264)
```

```
Takistipinge2 = Label(win, text = "U = " + format(ConvertVolts(voltage1_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

```
Takistipinge2.place(x=615,y=264)
```

```
Takistipinge3 = Label(win, text = "U = " + format(ConvertVolts(voltage2_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

```
Takistipinge3.place(x=827,y=264)
```

```
Takistipinge4 = Label(win, text = "U = " + format(ConvertVolts(voltage3_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

```
Takistipinge4.place(x=400,y=515)
```

```
Takistipinge5 = Label(win, text = "U = " + format(ConvertVolts(voltage4_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

```
Takistipinge5.place(x=615,y=515)
```

```
Takistipinge6 = Label(win, text = "U = " + format(ConvertVolts(voltage5_pärisväärtus,  
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
```

Lisa 1 järg

```
Takistipinge6.place(x=827,y=515)
Takistipinge7 = Label(win, text = "U = " + format(ConvertVolts(voltage_ch6_level, 1))
+ " V", font = myFont, bg = 'white', height = 1, width = 8)
Takistipinge7.place(x=400,y=765)
Takistipinge8 = Label(win, text = "U = " + format(ConvertVolts(voltage_ch7_level, 1))
+ " V", font = myFont, bg = 'white', height = 1, width = 8)
Takistipinge8.place(x=615,y=765)
Takistipinge9 = Label(win, text = "U = " + format(ConvertVolts(voltage_ch0_level2,
1)) + " V", font = myFont, bg = 'white', height = 1, width = 8)
Takistipinge9.place(x=827,y=765)
```

```
Takistivool1 = Label(win, text = "I = " +
format(ConvertAmperage(voltage0_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool1.place(x=400,y=286)
Takistivool2 = Label(win, text = "I = " +
format(ConvertAmperage(voltage1_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool2.place(x=615,y=286)
Takistivool3 = Label(win, text = "I = " +
format(ConvertAmperage(voltage2_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool3.place(x=827,y=286)
Takistivool4 = Label(win, text = "I = " +
format(ConvertAmperage(voltage3_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool4.place(x=400,y=537)
Takistivool5 = Label(win, text = "I = " +
format(ConvertAmperage(voltage4_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool5.place(x=615,y=537)
Takistivool6 = Label(win, text = "I = " +
format(ConvertAmperage(voltage5_pärisväärtus, 1)) + " A", font = myFont, bg = 'white',
height = 1, width = 8)
Takistivool6.place(x=827,y=537)
Takistivool7 = Label(win, text = "I = " + format(ConvertAmperage(voltage_ch6_level,
1)) + " A", font = myFont, bg = 'white', height = 1, width = 8)
Takistivool7.place(x=400,y=787)
Takistivool8 = Label(win, text = "I = " + format(ConvertAmperage(voltage_ch7_level,
1)) + " A", font = myFont, bg = 'white', height = 1, width = 8)
Takistivool8.place(x=615,y=787)
Takistivool9 = Label(win, text = "I = " + format(ConvertAmperage(voltage_ch0_level2,
1)) + " A", font = myFont, bg = 'white', height = 1, width = 8)
Takistivool9.place(x=827,y=787)
```

```
# If, Else Tingimus, Mis Kontrollib, Kas Vooluring On Lõplik
if (voolutugevus > 0.1):
    Kogutakistus = " R = " + format(kogutakistus) + " oomi"
```

Lisa 1 järg

```
else:
    Kogutakistus = "Vooluring pole lõplik"

# Kogutakistuse Ja Voolutugevuse Kuvamine
kogutakistus = Label(win, text = format(Kogutakistus), font = myFont, bg = 'white',
height = 2, width = 26)
kogutakistus.place(x=0,y=145)

kogutakistuslabel = Label(win, text = "Vooluringi Kogutakistus:", font = myFont, bg =
'white', height = 2, width = 26)
kogutakistuslabel.place(x=0,y=100)

voolutugevus = Label(win, text = "I = " + format(voolutugevus) + " A", font = myFont,
bg = 'white', height = 2, width = 24)
voolutugevus.place(x=0, y=445)

Voolutugevuslabel = Label(win, text = "Vooluringi voolutugevus:", font = myFont, bg =
'white', height = 2, width = 24)
Voolutugevuslabel.place(x=0, y=400)

# Tulemused Kuvatakse Iga Sekund
win.after(1000,GUImainloop)
win.after(1000,GUImainloop)

win.mainloop()
```


lisa 2. Lihtlitsents

Mina, _____ Klaid Hännile _____,
(*autori nimi*)

sünniaeg ____24.03.1999____,

1. annan Eesti Maaülikoolile tasuta loa (lihtlitsentsi) enda koostatud lõputöö

RASPBERRY PI-L PÕHINEV ÕPPEVAHEND VOOLURINGIDE
UURIMISEKS _____

(*lõputöö pealkiri*)

mille juhendaja(d) on _____ Heino Pihlap _____,
(*juhendaja(te) nimi*)

1.1. salvestamiseks säilitamise eesmärgil,

1.2. digiarhiivi DSpace lisamiseks ja

1.3. veebikeskkonnas üldsusele kättesaadavaks tegemiseks

kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile;

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Lõputöö autor _____ Klaid Hännile (allkirjastatud digitaalselt) _____
(*allkiri*)

Tartu, _____
(*kuupäev*)

Juhendaja(te) kinnitus lõputöö kaitsmisele lubamise kohta

Luban lõputöö kaitsmisele.

(*juhendaja nimi ja allkiri*)

(*kuupäev*)

(*juhendaja nimi ja allkiri*)

(*kuupäev*)